代数解析器及其密码学领域  
的应用与优化

杨博麟

摘要：本文对代数解析器进行相关介绍，并讨论其在密码学领域的应用以及可行的优化方案。作为自动化的可满足性问题的求解器，代数解析器可以被应用在许多领域，密码学的一些问题经过转化后也可以利用代数解析器进行求解。本文从两个角度讨论将这种求解进行优化的可能性。

# 绪论

## 可满足性问题

SAT(The propositional satisfiability problem, SAT)问题是逻辑学的一个基本问题，也是当今计算机科学和人工智能研究的核心问题。工程技术、军事、工商管理、交通运输及自然科学研究中的许多重要问题，如程控电话的自动交换、大型数据库的维护、大规模集成电路的自动布线、软件自动开发、机器人动作规划等，都可转化成SAT问题。可满足性SAT问题同时也是第一个被证明的NP-Complete问题，目前解决SAT问题已有的完全算法的运行时间呈指数增长。因此，对SAT算法加速的研究，具有很大的理论和应用价值，同时也有助于寻找优化其他复杂计算的方法。

在给出可满足性问题定义之前,首先介绍合取范式（Conjunctive Normal Form, CNF）合取范式是具有以下三个组成要素的布尔表达式：

字：一个布尔变量或一个布尔变量的取反：x1,¬x2

子句：一些字的析取（或）（布尔加）

合取范式(CNF)：一些子句的合取（与）构成的布尔公式

可满足性问题可以描述为： 给定一个布尔命题公式， 找到一组变量的赋值使得该命题公式为真（即，该公式可满足），或者证明该命题公式不可能为真（即，该公式不可满足）。可满足性问题是NP完全问题，因此研究可满足性问题在理论数学以及理论计算机方向有非常重要的理论价值，同时可满足性问题的求解具有极大的工业应用价值。电路可满足性问题(The circuit satisfiability problem )是最著名的NP完全问题之一，因为它是世界上第一个被证明的NP完全问题。而全解SAT问题，则是给定合取范式格式的布尔公式，判断是否存在一组变量赋值使得公式可满足。如果不存在，则该公式不满足。如果存在，则找出所有使得该公式满足的变量赋值。

# SAT问题的自动化求解

从求解方法来讲，基于冲突驱动、子句学习(the conflict-driven,clause-learning，简称CDCL)的方法是当今SAT求解中最流行求解范式。CDCL求解器通过对变量进行赋值来探索搜索空间，直到冲突出现。冲突分析将产生一个学习子句(learned clause),也叫冲突子句(conflict clause)，它表示这部分搜索空间不存在解，不应该再次被搜索。学习子句将被添加到学习子句数据库，合并到推理步骤。这个过程不断重复直到一个可满足的赋值被找到或者没有子句能够成为学习子句（即搜索完所有空间）。基础算法DPLL是在1962年首次被提出。这个算法是对早期DP(Davis-Putunam)算法的一个改进。

可满足性问题的自动求解框架主要可分为三类，分别是基于DPLL(Davis-Putnam-Logemann-Loveland)框架的求解器，基于组合(portfolio)的求解器以及端到端的基于深度神经网络的求解器。本文利用的CryptoMiniSAT求解器是基于DPLL框架的求解器。

简单来理解求解器的求解方法，其实是一种类似遍历求解但是效率又高于遍历的方法。其示意图如图1所示。

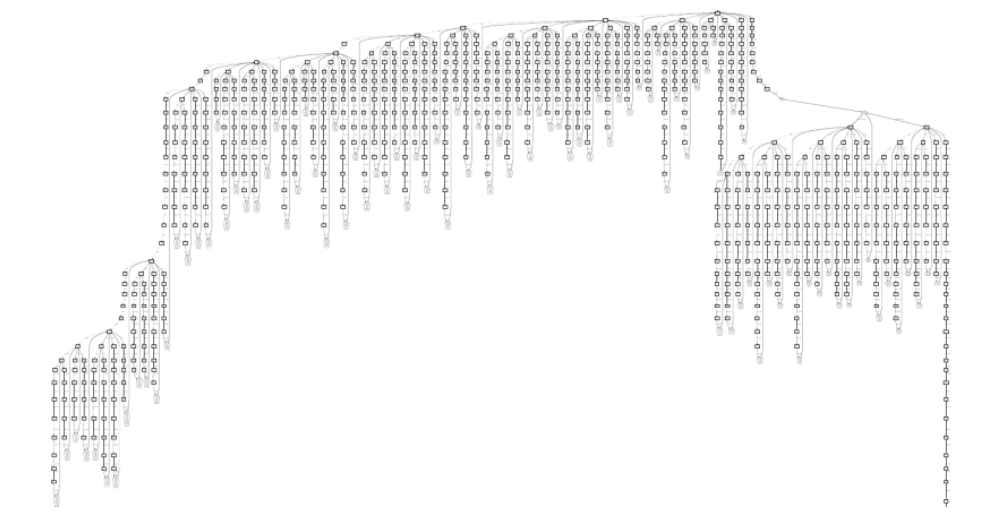


图 1 CryptoMiniSAT求解过程示意图

此求解方法与遍历法不同的一点是，基于上述提到的CDCL方法，在搜索过程中如果发现冲突，会把与此冲突相关的学习子句构成一个子空间，属于这个子空间内的候选值将不会被遍历。这与普通遍历相比大大缩小了搜索空间。

在代数解析器中，在求解SAT问题时，一般都将布尔代数式转化为CNF形式。示例如图2所示：

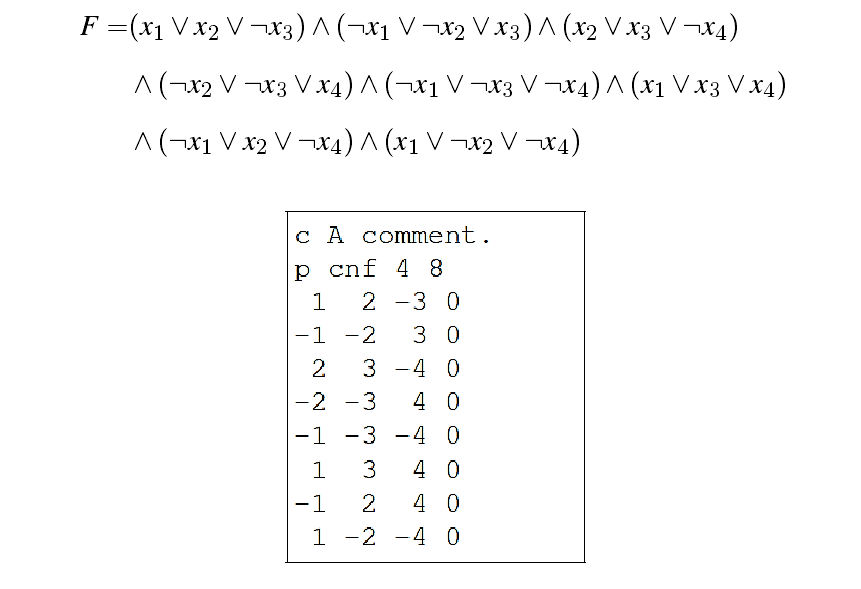


图 2 输入代数解析器的 CNF 形式

在CryptoMiniSAT求解器中，若子句由“x”开头，表明此子句为异或子句，其中的变量依次异或后结果为1，若在子句最后一个变量后加上负号“-”，则表示此子句实际取值为0。

## 密码学以及实现安全性

信息安全的重要性正随着全球信息化技术的不断发展而日益突显。各国都将信息安全作为国家策略进行重点研究。密码理论作为保障信息安全的主要工具，各种加密算法也在不断的升级与完善中。曾经人们对加密算法的安全性验证仅考虑算法自身被破解所需的计算复杂度，以此来判断算法是否安全。随着技术的发展，计算机的计算能力大幅度增高，加密算法为保障自身安全性，也不断提升计算复杂度以对抗。然而自从Kocher和Biham分别于1996年和1997年首次提出旁路密码分析和故障密码分析的思路后，加密算法在密码芯片上的实现安全性也进入了大众的考虑范围。

### 1.1.1旁路攻击

1996年Kocher首次提出旁路密码分析的思路，用差分计时的手段对密码算法进行分析以及密钥恢复。自此以后，旁路攻击受到学术界、军工界的高度关注。在二十几年的时间中，各界提出了许多新兴的攻击对象与分析手段。在一个硬件平台上如轻量级芯片、智能卡、FPGA上实现一个加密算法时，硬件平台中存在的成千上万个逻辑门电路共同运作。对应加密算法中的不同指令，逻辑门电路会采取不同的操作，会使硬件平台产生许多物理层面的泄露信息。从攻击对象角度来讲，可以针对芯片运行时间、功耗、电磁信号、芯片的Cache等来进行攻击。而从分析手段来讲，目前主流的旁路分析手段有：利用密码设备能量消耗的数据依赖性，对泄露信息进行差分运算的差分功耗分析（Differential Power Analysis: DPA）;利用密码设备能量消耗模型与处理数据间的相关系数进行分析的相关功耗分析（Correlation Power Analysis: CPA）;利用已知密码设备运行不同指令时的功耗模型不同进行分析的模板分析（Template Analysis: TA）以及结合了传统代数密码分析手段的代数旁路分析(Algebraic Side Channel Analysis：ASCA)等等。这些分析手段主要都是依据采集到的旁路信息，分析得到硬件平台上实现的加密算法的密钥等秘密信息。

2007年Bogdanov等首次将代数分析与碰撞旁路分析结合，针对AES，通过分析5条功耗曲线就恢复了完整密钥。2009年Renauld等提出了完整的代数旁路分析的思想，针对PRESENT以及AES进行密钥的恢复。

### 1.1.2故障攻击

上述提到的旁路分析手段都可算作被动攻击，是在密码芯片正确运行时采集信息并加以分析。而1997年Biham提出的故障攻击则可算作主动攻击。密码故障分析是一种利用电压毛刺、时钟毛刺、激光照射等手段，引起密码芯片运算模块故障运行的攻击方式。通过引入故障，并对引入故障后得到的错误密文等信息进行分析，最终达到恢复密钥或缩小密钥猜测空间大小的目的。自提出后，不同研究者针对不同算法进行了针对性的故障攻击并提出了相对的防御措施。2010年，Courtois将代数密码分析和故障分析结合，提出了代数故障攻击(Fault-Algebraic Attack);2018年，张帆等人提出了持续性故障分析（Persistent Fault Analysis）。

### 1.1.3代数密码分析

代数密码分析的提出，主要依据香农的理论：破解一个加密算法，工作量应与解决一个复杂的含大量未知数的联立方程组有同样多的工作。代数密码分析将整个加密算法转化为等价的代数方程组进行分析，而代数旁路分析则通过旁路分析手段将采集的旁路信息转化为同一类型的方程组，与加密算法的代数方程组进行联立，以达到在代数密码分析中降低方程计算复杂度的目的，同时也减少了旁路分析手段所需的样本数量，代数故障攻击的思路也大致相同。

PRESENT算法是一种轻量级的加密算法，由于其轻量级的特性常被用来测试攻击手段的可行性。该算法的提出者在文中证明，PRESENT算法可用4216个未知变量和11067个二次方程构成的代数方程组表示，将S盒代换用8个GF(2)上的变量，通过21个二次方程来表示。并证明利用现有的代数化方法无法进行有效的代数攻击。2009年Renauld等人首次针对PRESENT算法应用代数旁路攻击；2010年卜凡等人对低轮PRESENT进行了代数攻击；文献首次对PRESENT加密算法进行差分故障分析，将主密钥的搜索空间降低至216；赵新杰等人基于故障的传播路径，8次故障注入即可将PRESENT-80密钥的搜索空间降低至214.7；2012年吴克辉等人首次针对PRESENT提出代数故障攻击，可在50s内恢复64比特最后一轮白化密钥，最少仅需两次故障注入。

由于针对的是完整的算法的等效方程组。若单纯考虑旁路攻击或故障攻击，则是分别对方程组的前半部分或后半部分的解的空间进行了限制。如果能将两者结合，那么整体方程组的求解难度将再次降低。同时根据上述国内外研究进展，旁路攻击和故障攻击在某些方面可以互相促进攻击效率，如使用旁路信息辅助确定故障注入的轮数，以便于后续分析等。因此我们考虑从以上两个角度进行探索性研究。